



Критические ситуации в рекурсивном решении

Разберём основные проблемы, возникающие при рекурсивных вычислениях, и способы их устранения.

Неверный ввод n (отрицательное число)

Причина: Отрицательное число или 0. В рекурсии может привести к бесконечным вызовам.

Решение: Использовать `unsigned int` вместо `int` — отрицательные числа отсекаются автоматически. Добавить проверку `if (n == 0) return 1`

Переполнение рекурсии (Stack Overflow)

Причина: Слишком большая глубина рекурсии при больших n .

Решение: Заменить рекурсию на обычный цикл `for` или `while`. Это полностью убирает риск переполнения стека.

Арифметические ошибки и некорректный ввод

Переполнение результата (арифметическое)

Причина: long long переполняется уже на **21!**

Решение: Использовать long double — он выдержит до **1000!** без переполнения.

Ввод мусора / не числа

Причина: cin >> n не работает, n останется мусорным значением.

Решение: Проверить ввод:

```
if (!(cin >> n)) {  
    cout << "Ошибка! n = 1";  
    n = 1;  
    cin.clear();  
}
```

